CERN-TH/95-169

SABSPV – A Monte Carlo integrator for Small-Angle Bhabha Scattering

Matteo CACCIARI [1]
DESY - Deutsches Elektronen-Synchrotron
Notkestrasse 85 - 22607 HAMBURG - GERMANY

Guido MONTAGNA
Dipartimento di Fisica Nucleare e Teorica, Università di Pavia
Via A. Bassi n. 6 - 27100 PAVIA - ITALY

Oreste NICROSINI [2]
CERN, Theory Division
CH 1211 - Geneva 23 - SWITZERLAND

Fulvio PICCININI
Istituto Nazionale di Fisica Nucleare, Sezione di Pavia
Via A. Bassi n. 6 - 27100 PAVIA - ITALY

Program classification: 11.1

SABSPV is a code designed to perform a theoretical evaluation of small-angle Bhabha scattering cross sections by suitably matching fixed-order perturbative calculations and structure-function techniques. The implementation of realistic experimental triggering conditions is achieved by using Monte Carlo integration.

Submitted to Computer Physics Communications

CERN-TH/95-169
July 1995

[1] Della Riccia and Università di Pavia Fellow. Work supported by INFN, Sezione di Pavia, Italy.

[2] On leave from INFN, Sezione di Pavia, Italy.

PROGRAM SUMMARY

Title of program: SABSPV

Computer: DEC VAX, HP/APOLLO, IBM/RS6000; *Installation:* INFN, Sezione di Pavia, via A. Bassi 6, 27100 Pavia, Italy

Operating system: VMS, UNIX

Programming language used: FORTRAN 77

Memory required to execute with typical data: 40 kByte

No. of bits in a word: 32

The code has not been vectorized

Subprograms used: random number generator RANLUX [1]; routines from the CERN Program Library are also used.

No. of lines in distributed program, including test data etc.: 1700.

Correspondence to:
CACCIARI@DESY.DE, CACCIARI@PV.INFN.IT;
MONTAGNA@PV.INFN.IT;
NICROSINI@VXCERN.CERN.CH, NICROSINI@PV.INFN.IT;
PICCININI@PV.INFN.IT

Keywords: high energy electron–positron collisions, small-angle Bhabha scattering, luminosity determination, QED corrections, electron structure functions, experimental cuts, Monte Carlo integration.

Nature of physical problem
The precise determination of the theoretical Bhabha scattering cross section in the small-angle regime is a key ingredient for precision luminometry at LEP. To this aim, QED radiative corrections to the tree-level Standard Model Cross section have to be taken into account, together with vacuum polarization effects. Particular care has to be devoted to higher-order corrections. The theoretical formulation must allow the computation of the cross section for a wide variety of cuts on the final-state particles.

Method of solution
A suitable matching of a fixed-order calculation with the structure-function techniques for resumming large initial- and final-state leading logarithmic corrections [2] is performed. A Monte Carlo integration with weighted events has been implemented in order to mimic as close as possible the experimental triggering conditions. The importance-sampling technique [3] is employed to take care of the peaking behaviour of the integrand.

Restrictions on the complexity of the problem
The $\mathcal{O}(\alpha)$ QED corrections are computed exactly only for the dominant contribution to the small-angle cross section, namely the square of γ-exchange in the t-channel; all the other contributions to the cross sections are corrected at the leading-logarithmic level. The contribution of additional hadronic or leptonic pairs is at present neglected. Starting from $\mathcal{O}(\alpha^2)$, QED corrections are implemented at the leading-logarithmic level.

Typical running time
On a HP 9000/735 the code takes about 2×10^{-4} seconds per event, with standard cuts. A one per mille relative error can be achieved with about 10^7 events.

Unusual features of the program: none

References

[1] F. James, Comput. Phys. Commun. 79 (1994) 111.

[2] M. Cacciari, G. Montagna, O. Nicrosini and F. Piccinini, in "Reports of the Working Group on Precision Calculations for the Z Resonance", eds. D. Bardin, W. Hollik and G. Passarino (CERN Report 95-03, Geneva, 1995), p. 389, and references therein.

[3] F. James, Rep. Prog. Phys. 34 (1980) 1145.

LONG WRITE – UP

1 Introduction

Precision luminometry at LEP requires the best possible knowledge of the theoretical Bhabha scattering cross section in the small-angle regime. Actually, the experimental precision of luminosity measurements is at present at the level of 0.1% and is foreseen to improve further in the near future. In order to exploit such an experimental precision, theoretical results of comparable accuracy are required.

Two kinds of problems are to be faced here. The first one is to establish the physical precision of a given theoretical approach; this requires an estimate of those contributions to the cross section which are neglected by the approach itself. The second one is to establish the technical precision of the computational realization of a given theoretical approach; this requires an estimate of the errors introduced by numerical approximations, different implementations of theoretically known effects and so on, and, last but not least, bugs in the computer code.

The comparison between different and independent approaches can contribute to the solution of both problems and become important in order, on the one hand, to assess the presently reached accuracy and, on the other, to eventually improve it in a significant way.

Some codes that compute the small-angle Bhabha scattering cross section already exist [1, 2]. The aim of the present work is to describe in some detail the code `SABSPV` .

In order to compute a small-angle Bhabha scattering cross section with a precision of the order of 0.1% and within realistic experimental cuts, the following tasks have to be accomplished:

1. to use a theoretical formulation capable of taking into account the effects of multiple soft and/or collinear photon emission from initial- and final-state particles, but also of providing an exact evaluation of at least one hard photon emission;

2. to perform an implementation of the integration within the phase-space limits such that a wide variety of experimental triggering conditions can be closely reproduced.

The first point is solved by developing a formulation, described in [3], which allows the matching of a fixed-order perturbative calculation with the structure-function technique. The outcome of this matching is a calculation which correctly describes one hard photon emission but also allows for the evaluation of multiphoton emission in the initial and final state.

As for the second point, the cuts are implemented by using a Monte Carlo integration. The four-momenta of the outgoing particles (electron, positron and photon) are constructed and fed to a cutting routine that reproduces the experimental triggering conditions.

In the following we will first review the theoretical background and its implementation in the code. The structure of the integrand demands the importance-sampling technique to be applied in order to have an efficient Monte Carlo integration. We will discuss the way this has been done. Finally, the program structure will be briefly presented and the required inputs and the corresponding output will be described.

2 Theoretical Background

The structure-function approach allows the leading-log-corrected cross section in the laboratory frame to be written, taking into account all-orders photonic radiation, in the following form:

$$\begin{aligned}\sigma_{LL}^{(\infty)} &= \int d\Omega dx_1 dx_2 D^{(\infty)}(x_1, Q^2) D^{(\infty)}(x_2, Q^2) J(x_1, x_2, \vartheta) \\ &\times \frac{d\sigma}{d\Omega}(\hat{s}(x_1, x_2), \hat{t}(x_1, x_2, \vartheta)) F^{(\infty)}(\xi_1, Q^2) F^{(\infty)}(\xi_2, Q^2) \Theta(\text{cuts}). \qquad (1)\end{aligned}$$

The detailed description of Eq. (1) can be found in [3, 4]. For convenience, we recall that σ is the Born cross section, the D's are the structure functions for initial-state radiation and the F's describe final-state photonic emission; $\Theta(\text{cuts})$ represents the rejection algorithm for implementing the experimental cuts.

It was prompted in the Introduction that when aiming at a per mille accuracy we have to rely also on a complete $\mathcal{O}(\alpha)$ calculation of small-angle Bhabha scattering cross section. The complete squared matrix element of the (gauge-invariant) t-channel diagrams associated with (hard) photon radiation from the electron line has been computed in [5]. Its full expression, including all $\mathcal{O}(m^2)$, $\mathcal{O}(m^4)$ and $\mathcal{O}(m^6)$ terms, reads as follows:

$$|M|^2 = \frac{2(4\pi\alpha)^3}{t^2}\left\{M_0 + m^2 M_2 + m^4 M_4 + m^6 M_6\right\}, \qquad (2)$$

where the single terms M_i are explicitly reported in [5]. From this matrix element the $\mathcal{O}(\alpha)$ cross section can be computed via Monte Carlo integration as

$$\sigma_\gamma^H(k_0) = \int d[PS]\; F\Big(|M_-|^2 + |M_+|^2\Big)\Theta(\text{cuts}), \qquad (3)$$

where $d[PS]$ is the phase-space volume element, $|M_\pm|^2$ are the squared amplitudes for the electronic and positronic radiation, F is the proper flux factor, and k_0 is a minimum energy fraction of the radiated photon, defined as $k_0 = E_{min}^\gamma / E$.

The matching between the all-orders leading-log cross section of eq. (1) and the $\mathcal{O}(\alpha)$ one (3) takes place in the following way: the order-α content of the leading-log t-channel γ-exchange cross section is extracted by employing the $\mathcal{O}(\alpha)$ expansions of the structure functions entering the master formula (1), thereby yielding $\sigma_{LL,\gamma}^{(\alpha)}$. Denoting by $\sigma_\gamma^{S+V}(k_0)$ the t-channel γ-exchange cross section including virtual corrections plus soft photons of energy up to $E_\gamma = k_0 E$ (see for instance [6]), the cross section in the luminometry region can finally be written as

$$\sigma_A = \sigma_{LL}^{(\infty)} - \sigma_{LL,\gamma}^{(\alpha)} + \sigma_\gamma^{S+V}(k_0) + \sigma_\gamma^H(k_0), \qquad (4)$$

where, we recall, $\sigma_{LL}^{(\infty)}$ is the all-orders leading-log-corrected full Bhabha cross section of Eq. (1), $\sigma_{LL,\gamma}^{(\alpha)}$ is the up to $\mathcal{O}(\alpha)$ leading-log-corrected cross section, limited to the t-channel γ-exchange contribution, and $\sigma_\gamma^H(k_0)$ is the radiative Bhabha cross section of Eq. (3).

The difference $\sigma_{LL}^{(\infty)} - \sigma_{LL,\gamma}^{(\alpha)}$ contains the Born contribution of all the Bhabha channels but the t-channel γ-exchange one, plus their leading-log QED corrections resummed to all orders, plus the higher-order leading-log QED corrections to the t-channel γ-exchange contribution starting from $\mathcal{O}(\alpha^2)$; the exact up to $\mathcal{O}(\alpha)$ contribution for the t-channel γ-exchange term is then supplied by $\sigma_\gamma^{S+V}(k_0)+\sigma_\gamma^H(k_0)$.

Equation (4) is in the additive form. A factorized form can also be supplied. It has the same $\mathcal{O}(\alpha)$ content but also leads to the so-called classical limit, according to which the cross section must vanish in the absence of photonic radiation. It reads

$$\sigma_F = (1+C_{NL}^H)\sigma_{LL}^{(\infty)},$$

$$C_{NL}^H \equiv \frac{\sigma_\gamma^{S+V}(k_0)+\sigma_\gamma^H(k_0)-\sigma_{LL,\gamma}^{(\alpha)}}{\sigma} \equiv \frac{\sigma_{NL,\gamma}^{(\alpha)}}{\sigma}, \tag{5}$$

σ being the Born cross section; C_{NL}^H contains the non-log part of the $\mathcal{O}(\alpha)$ $\gamma(t)\gamma(t)$ cross section, represented by $\sigma_{NL,\gamma}^{(\alpha)}$. A detailed discussion on the physical content of eqs. (4) and (5) will be given elsewhere.

3 Importance Sampling

Both the leading-log cross section and the real hard photon one present singularities which have to be properly treated with importance-sampling techniques to ensure a reasonably fast convergence of the Monte Carlo integration. This means dividing the integrand by a so-called weight function and redefining the integration variable:

$$\int f(x)dx = \int \frac{f(x)}{p(x)}p(x)dx \rightarrow \int \frac{f(y)}{p(y)}dy \tag{6}$$

with $y=\int p(x)dx$. The weight function $p(x)$ must be chosen such that the ratio $f(x)/p(x)$ is smooth enough to produce a small variance.

Let us start with the leading-log cross section. The structure function $D^{(\infty)}(x,Q^2)$ reads

$$\begin{aligned} D^{(\infty)}(x,Q^2) &= \frac{\exp\left\{\frac{1}{2}\beta(\frac{3}{4}-\gamma_E)\right\}}{\Gamma(1+\frac{1}{2}\beta)}\frac{1}{2}\beta(1-x)^{\frac{1}{2}\beta-1} - \frac{1}{4}\beta(1+x) \\ &+\frac{1}{32}\beta^2\Big[-4(1+x)\ln(1-x)+3(1+x)\ln x \\ &-4\frac{\ln x}{1-x}-5-x\Big], \qquad \beta=\frac{2\alpha}{\pi}\left[\ln\left(\frac{Q^2}{m_e^2}\right)-1\right], \end{aligned} \tag{7}$$

and displays a singularity in $x=1$. This is disposed of by employing, in connection with each structure function in the integrand, a weight function that resembles the singular part of the structure function itself:

$$p(x) = \frac{1}{2}\beta(1-x)^{\frac{1}{2}\beta-1}. \tag{8}$$

The relation between the original integration variable x and the flat random variable $r\in(0,1)$ is then

$$x = 1-(1-r)^{2/\beta}. \tag{9}$$

The real hard photon cross section evaluation must instead face the “singular”(in fact, only extremely peaked) distribution of the relative angle θ_γ between the photon and the emitting particle when approaching the forward collinear region. Four such "singularities" are present, related to initial- and final-state emission from electron and positron respectively. The initial-state ones are smoothed by choosing the weight function

$$p(\cos\theta_\gamma) = \left(1 - \sqrt{1 - m_e^2/E^2}\,\cos\theta_\gamma\right)^{-1}, \tag{10}$$

m_e and E being the electron mass and energy in the lab frame, respectively. The final-state ones are disposed of by using similar weight functions after a proper rotation of the reference frame such that the z-axis is taken along the outgoing lepton.

4 Program Structure

The code has the usual structure of Monte Carlo programs, with routines evaluating the cross sections (some pertaining to the real hard photon part, some to the leading-log cross section), triggering routines, determining whether the event is to be accepted or not, utility routines, and finally routines collecting the output. More in detail, we have:

- Monte Carlo routines:
  - the `MAIN PROGRAM`: where the Monte Carlo loop is located. It provides the random numbers to the routine `INTEGR`, collects its outputs, performs the statistics operations and writes the results to the output file;
  - `SUBROUTINE INTEGR`: receives the random numbers from the `MAIN PROGRAM` and calls the cross sections, kinematics and triggering routines. It returns the cross sections for a given event to the `MAIN PROGRAM`.
- Cross section routines:
  - `SUBROUTINE HARD`, `SUBROUTINE EEG`: provide the real hard photon cross section;
  - `FUNCTION BHABORN`, `FUNCTION BHABORN_gtgt`, `FUNCTION AINTSV`: provide the full Born cross section, the Born cross section for the $\gamma(t)\gamma(t)$ channel and the virtual plus soft $\mathcal{O}(\alpha)$ correction to the latter, respectively;
  - `FUNCTION DOP`, `FUNCTION AJAC`, `FUNCTION PDINFTY`, `FUNCTION PDPDALPHA`: routines related to the structure-function evaluation of the leading-log cross section;
  - `FUNCTION VPOL`, `FUNCTION SOMMAT`, `FUNCTION ABC`: evaluate the vacuum polarization correction to the photon propagator, according to the parametrization of [7].
- Kinematic routines:
  - `SUBROUTINE KINE2`, `SUBROUTINE QUADRI2`: build up the outgoing particles four-momenta from the generated random numbers for two-body events, i.e. Born and virtual plus soft cross sections, and collinear leading-log ones;
  - `SUBROUTINE KINEH`, `SUBROUTINE QUADRIH`: the same but for three-body events, with the real hard photon.

- Triggering routines:
  - SUBROUTINE CUTS: master cut routine. It propagates to the rest of the code the result of one of the triggering routines;
  - SUBROUTINE TRIGGER1: performs acceptance cuts for a sample trigger;
  - SUBROUTINE TRIGGER2: dummy routine to be used to implement a new trigger.
- Utility routines:
  - FUNCTION EXCHANGE, FUNCTION COSANGLE, FUNCTION PHIROT: they exchange any two variables, evaluate the cosine of the angle between any two four-momenta and rotate a four-momentum around its z-axis, respectively.

The code can be used in a "black box" way with the only exception of the triggering routine, which may be provided by the user. An example of how to build such a routine is given in Section 5. We now want to describe the common blocks the user has to be familiar with for writing his own triggering routine.

-
  ```
        REAL*8 QUADRIM(0:3), QUADRIP(0:3), KUADRI(0:3)
        COMMON/MOMENTA/QUADRIM,QUADRIP,KUADRI
  ```

  This common block contains the four-momenta of the outgoing electron (QUADRIM), positron (QUADRIP) and photon (KUADRI). It is filled by the QUADRI2 or QUADRIH routines when treating soft-collinear or real hard photons respectively.

-
  ```
        COMMON/CUTOFF/EGMIN,IHARD
  ```

  These are non-physical parameters. EGMIN contains the value of the cutoff k_0, which separates virtual plus soft and real hard photon events. IHARD is a flag which is set to 1 when processing real hard photon events, and 0 otherwise. It can be used in the triggering routines to skip or implement some cuts according to the event being of two-body/collinear type or with a real hard photon. For example, the cut on the k_0 cutoff only has to be activated in the real hard photon part, being taken into account analytically in the virtual plus soft part.

-
  ```
        COMMON/CONST/EBEAM,ALPHA,CONVFAC,PI,AME
  ```

  Some constants, which are used in many points of the code. EBEAM is the beam energy in GeV (see Section 6 about input parameters). ALPHA is the electromagnetic coupling, $\alpha = 1/137.0359895$. CONVFAC is the conversion factor from GeV^{-2} to nb: $.38937966 \times 10^6$. Pi is π and, finally, AME is the electron mass: $m_e = 0.51099906 \times 10^{-3}$ GeV.

-
  ```
        COMMON/EXPCUTS/T1MIN,T1MAX,T2MIN,T2MAX,E1MIN,E2MIN
  ```

  The experimental acceptance cuts. See Section 6 about inputs description.

-
  ```
        REAL*8 CALOINPUT(5)
        COMMON/CALOS/CALOINPUT
  ```

  These are parameters which are read in from the input file (once more, see Section 6) and which can be used in writing the triggering routines. The meaning of the five parameters will depend on the use which is made of them in the routines. For instance, in TRIGGER1 the parameter CALOINPUT(1) is the calorimetric threshold as a fraction of the beam energy and CALOINPUT(2) is the half-opening (in radians) of the cone, which defines an electromagnetic cluster.

5 Sample Trigger Routine

A sample triggering routine TRIGGER1() is inserted in the code. It can be used as an example to construct other experimental-like triggers. A dummy TRIGGER2() is also given, to be completed by the user.

Below we give the listing of TRIGGER1(). It first checks for the electron and the positron to be within angular acceptance cuts. After that, it checks whether one of the leptons and the photon do form an electromagnetic cluster, i.e. are radially separated by less than the critical distance fixed by the input parameter DEL. If they do, the energy cut is performed using the sum of their energies rather than the energy of the bare fermion only.

```
      LOGICAL FUNCTION TRIGGER1()
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 QUADRIM(0:3), QUADRIP(0:3), KUADRI(0:3)
      REAL*8 ZETA(0:3)
      REAL*8 CALOINPUT(5)
      COMMON/MOMENTA/QUADRIM,QUADRIP,KUADRI
      COMMON/EXPCUTS/T1MIN,T1MAX,T2MIN,T2MAX,E1MIN,E2MIN
      COMMON/CUTOFF/EGMIN,IHARD
      COMMON/CONST/EBEAM,ALPHA,CONVFAC,PI,AME
      COMMON/CALOS/CALOINPUT
      DATA ZETA/0D0,0D0,0D0,1D0/
*
      EGAMMA = KUADRI(0)
*
      IF(IHARD.EQ.1) THEN
*
*.....THIS IS THE NON-PHYSICAL CUTOFF
*
        IF(EGAMMA.LT.EGMIN) THEN
           TRIGGER1 = .FALSE.
           RETURN
        ENDIF
      ENDIF
*
      THETA1 = ACOS(COSANGLE(QUADRIM,ZETA))
      THETA2 = ACOS(COSANGLE(QUADRIP,ZETA))
      THETA2OPP = PI-THETA2
      E1 = QUADRIM(0)
      E2 = QUADRIP(0)
*
      IF(THETA1.LT.T1MIN) THEN
         TRIGGER1 = .FALSE.
         RETURN
      ENDIF
*
      IF(THETA1.GT.T1MAX) THEN
         TRIGGER1 = .FALSE.
         RETURN
      ENDIF
*
      IF(THETA2OPP.LT.T2MIN) THEN
         TRIGGER1 = .FALSE.
         RETURN
      ENDIF
*
      IF(THETA2OPP.GT.T2MAX) THEN
         TRIGGER1 = .FALSE.
         RETURN
      ENDIF
*
      EMCL = E1
      EPCL = E2
*
      IF (IHARD.EQ.1) THEN
        DJET = CALOINPUT(2)
        CDJET = COS(DJET)
        CEG = COSANGLE(QUADRIM, KUADRI)
        CPG = COSANGLE(QUADRIP, KUADRI)
*
        IF(CEG.GE.CDJET) THEN
           EMCL = E1 + EGAMMA
        ENDIF
*
        IF(CPG.GE.CDJET) THEN
           EPCL = E2 + EGAMMA
        ENDIF
*
      ENDIF
*
      CALOTH = CALOINPUT(1)
      SUMEN = EMCL*EPCL/EBEAM/EBEAM
*
      IF(SUMEN.LT.CALOTH) THEN
         TRIGGER1 = .FALSE.
         RETURN
      ENDIF
*
      TRIGGER1 = .TRUE.
*
      END
```

6 Input Description

A data card of the following kind has to be provided, via standard input, when running the program:

```
46.15D0                                ! EBEAM
24.D-3  58.D-3  0.D0                   ! T1MIN, T1MAX, E1MIN
24.D-3  58.D-3  0.D0                   ! T2MIN, T2MAX, E2MIN
0.5D0   1.D-2   0.D0  0.D0  0.D0       ! CALOINPUT(1...5)
1                                      ! ISIM
1                                      ! ICALO
1.D5    0.D0    0     'SABSPV.OUT'     ! EVTS, ACCLIM, IRESTART, OUTFILE
```

These parameters have the following meaning.

First line: `46.15D0` - the electron and positron beam energy, `EBEAM`.

Second line: `24.D-3, 58.D-3, 0.D0` - the electron minimum and maximum scattering angle (in radians) and the minimum visible energy (in GeV), `T1MIN, T1MAX, E1MIN`. These cuts are to be interpreted as "fiducial" cuts within which the events are generated, before going through the triggering routine.

Third line: the same for the positron, `T2MIN, T2MAX, E2MIN`.

Fourth line: `0.5D0, 1.D-2, 0.D0, 0.D0, 0.D0` - inputs that may be required by the cutting routines for the triggers. These value are stored in the vector `CALOINPUT(5)` via the common block `COMMON/CALOS`. In this particular input the first figure will represent the calorimetric threshold and the second one the half-opening angle of the electromagnetic jet, while the other three will go unused.

Fifth line: `1` - flag for symmetric cuts, `ISIM`. The user has to specify if the experimental cuts asked for are (`1`) or not (`0`) symmetric for electron–positron exchange. If they are, choosing `1` saves computing time.

Sixth line: `1` - flag for choosing the triggering routine, `ICALO`. Possible choices are:

1. - `TRIGGER1` (sample trigger)
2. - `TRIGGER2` (free slot)

Seventh line: `1.D4, 0.D0, 0, 'SABSPV.OUT'` - these are inputs related to the Monte Carlo integration and to the management of the output. Namely, the total number of events, `EVTS`, the relative accuracy limit aimed at (the program stops when the statistical error reaches this level), `ACCLIM`, the restarting flag, `IRESTART` (if `1` the program tries to restart execution from the indicated output file, if `0` it reinitializes it), and the output file name, `OUTFILE`.

7 Test Run Output

The input file given in Section 6 produces the following output:

```
 CM ENERGY      =   92.30000

 THETA_EL_MIN   =     .02400
 THETA_EL_MAX   =     .05800
 THETA_POS_MIN  =     .02400
 THETA_POS_MAX  =     .05800
 EN_EL_MIN      =     .00051
 EN_POS_MIN     =     .00051
 CALO INPUTS    =     .50000     .01000     .00000     .00000     .00000
```

```
SIMM. FLAG     =          1

TRIGGER CHOICE =    TRIGGER1

CUTOFF         =     .0004758500


FULL (LL)             =    174.34870 +/-   .35262
O(ALPHA) (GTGT)       =     -8.85671 +/-   .06548
BORN PART (GTGT)      =    182.98221 +/-   .29020
H.O. GTGT AND LL 2,8  =       .22320 +/-   .21581
VIRTUAL + SOFT (GTGT) =    -91.12947 +/-   .18056
HARD PHOTON (GTGT)    =    263.66676 +/-   .65744
ORDER ALPHA (GTGT)    =    172.53729 +/-   .70125
COMPLETE X-SECT       =    172.76049 +/-   .76344
COMPL. X-SECT (FACT)  =    172.83542 +/-  1.07296

NUMBER OF EVENTS:     100000
RANLUX INITIAL SEQUENCE:      4              1              0              0
RANLUX RESTARTING SEQUENCE:   4              1        8104045              0

SUM VECTOR:
        .1743487004930795E+08
       -.8856709767153972E+06
        .1829822083556865E+08
        .2232019045425988E+05
       -.9112947344238646E+07
        .2636667600328065E+08
        .1727604884949618E+08
        .1725372865904168E+08

SUM^2 VECTOR:
        .4283122616525836E+10
        .5072615600831745E+08
        .4190405492369947E+10
        .4657244067723529E+09
        .1156479410248304E+10
        .1127428205612726E+11
        .8812978799192981E+10
        .7894492407254981E+10
```

References

[1] S. Jadach, E. Ricther-Was, B.F.L. Ward and Z. Was, Comput. Phys. Commun. **70** (1992) 305; Phys. Lett. **B260** (1991) 438; Phys. Lett. **B268** (1991) 253.

[2] M. Caffo, H. Czyź and E. Remiddi, Phys. Lett. **B327** (1994) 369; Int. J. Mod. Phys. **4** (1993) 591; Nuovo Cimento **105A** (1992) 277.

[3] M. Cacciari, G. Montagna, O. Nicrosini and F. Piccinini, in "Reports of the Working Group on Precision Calculations for the Z Resonance", eds. D. Bardin, W. Hollik and G. Passarino (CERN Report 95-03, Geneva, 1995), p. 389, and references therein.

[4] G. Montagna, O. Nicrosini and F. Piccinini, Phys. Rev. **D48** (1993) 1021.

[5] M. Greco, G. Montagna, O. Nicrosini and F. Piccinini, Phys. Lett. **B318** (1993) 635; G. Montagna, O. Nicrosini and F. Piccinini, Comput. Phys. Commun. **78** (1993) 155; erratum, ibid. **79** (1994) 351.

[6] M. Caffo and E. Remiddi, *Bhabha Scattering*, CERN Report 89–08 Vol. 1 (1989) 171, G. Altarelli, R. Kleiss and C. Verzegnassi Eds., Geneva 1989, and references therein.

[7] H. Burkhardt, F. Jegerlehner, G. Penso and C. Verzegnassi, Z. Phys. **C43** (1989) 497.